



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|-----------------|-------------|----------------------|---------------------|------------------|
| 10/083,872 | 02/27/2002 | Eric DeLano | 10016663-1 | 4721 |

7590 09/20/2007
HEWLETT-PACKARD COMPANY
Intellectual Property Administration
P.O. Box 272400
Fort Collins, CO 80527-2400

| |
|----------|
| EXAMINER |
|----------|

LI, AIMEE J

| | |
|----------|--------------|
| ART UNIT | PAPER NUMBER |
|----------|--------------|

2183

| | |
|-----------|---------------|
| MAIL DATE | DELIVERY MODE |
|-----------|---------------|

09/20/2007

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

| | | | |
|------------------------------|--------------------------------------|-------------------------------------|--|
| Office Action Summary | Application No. 10/083,872 | Applicant(s) DELANO, ERIC | |
| | Examiner Aimee J. Li | Art Unit 2183 | |

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 09 July 2007.
 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 19-23, 27-29 and 31-33 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) ☐ Claim(s) _____ is/are allowed.
 6) ☒ Claim(s) 19-23, 27-29 and 31-33 is/are rejected.
 7) ☐ Claim(s) _____ is/are objected to.
 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
 10) ☒ The drawing(s) filed on 04 April 2005 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) ☐ All b) ☐ Some * c) ☐ None of:
 1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date <u>4/16/07</u> . | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. Claims 19-23, 27-29, and 31-33 have been considered. Claims 24-26, 30, and 34 have been cancelled as per Applicants' request. Claims 19, 23, 27-28, 31, and 33 have been amended as per Applicants' request.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: Amendment as filed 09 July 2007.

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

4. Claims 19-23, 27, 29, and 31-32 are rejected under 35 U.S.C. 102(e) as being taught by Fernando et al., U.S. Patent Number 6,272,616 (herein referred to as Fernando).

5. Referring to claim 19, Fernando has taught a method for processing bundled instructions through execution units of a processor, comprising the steps of:

- a. Determining a mode of operation, wherein the mode of operation comprises one of a throughput mode and a wide mode (Fernando column 2, lines 51-62 "...The architecture is capable of running in various modes, including single threaded mode, SIMD mode and MIM mode..."; column 3, line 29-33 "...at least three different modes of parallelism..."; column 6, lines 33-40 "...CFORK instruction

is one of the special instructions and, specifically, is the instruction which activates one or more secondary instruction pipelines in the SIMD mode...”; column 7, lines 14-35 “...DFORK is one of the special instructions. This instruction indicates that the architecture is to enter the MIMD mode...”; and Figure 5);

b. In the throughput mode:

- i. Fetching a first bundle of instructions from a first thread of a multiply threaded program (Fernando column 3, line 62-67 “In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data...”; column 7, lines 6-46 “...The two threads 16 and 18 then will operate essentially independently of each other, reading and executing different instructions and using different data...”; column 11, line 37 to column 12, line 4 “...a DFORK instruction causing instruction pipeline 2 to being operating in MIMD mode at time t_2 ...”; Figure 4; Figure 5; and Figure 7);
- ii. Distributing the first bundle to a first cluster of the execution units for execution therethrough (Fernando column 3, line 62-67 “In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and

fetches its own data...”; column 7, lines 6-46 “...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data...”; column 11, line 37 to column 12, line 4 “...a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time **t2**...”; Figure 4; Figure 5; and Figure 7);

- iii. Fetching a second bundle of instructions from a second thread of the program (Fernando column 3, line 62-67 “In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data...”; column 7, lines 6-46 “...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data...”; column 11, line 37 to column 12, line 4 “...a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time **t2**...”; Figure 4; Figure 5; and Figure 7 – In regards to Fernando, in MIMD mode, each instruction pipeline works independent of the others to fetch instructions from the individual threads associated with the instruction pipeline and to execute those instructions.);
- iv. Distributing the second bundle to the first cluster of the execution units for execution therethrough (Fernando column 3, line 62-67 “In the MIMD

mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data...”; column 7, lines 6-46 “...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data...”; column 11, line 37 to column 12, line 4 “...a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time **t2**...”; Figure 4; Figure 5; and Figure 7);

- v. Fetching a third bundle of instructions from a third thread of the program (Fernando column 3, line 62-67 “In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data...”; column 7, lines 6-46 “...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data...”; column 11, line 37 to column 12, line 4 “...a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time **t2**...”; Figure 4; Figure 5; and Figure 7 – In regards to Fernando, Fernando teaches in column 5, line 54 to column 6, line 3 that portions of the program benefit from the use of MIMD and SIMD modes to improve processing performance and speed, meaning that using

the DFORK instruction would improve performance in certain portions of the code, so using the DFORK instruction in all possible instances to fetch instructions from new threads is taught.);

- vi. Distributing the third bundle to a second cluster for execution therethrough (Fernando column 3, line 62-67 “In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data...”; column 7, lines 6-46 “...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data...”; column 11, line 37 to column 12, line 4 “...a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time **t2**...”; Figure 4; Figure 5; and Figure 7);
- vii. Fetching a fourth bundle of instructions from a fourth thread of the program (Fernando column 3, line 62-67 “In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data...”; column 7, lines 6-46 “...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data...”; column 11, line 37 to column 12, line 4 “...a DFORK instruction causing instruction pipeline **2**

to being operating in MIMD mode at time t_2 ...”; Figure 4; Figure 5; and Figure 7); and

- viii. Distributing the fourth bundle to the second cluster for execution therethrough (Fernando column 3, line 62-67 “In the MIMD mode, the full hardware of two or more of the instruction pipelines are fully operations...each instruction pipeline individually fetches instructions from the instruction memory using its own fetch stage and fetches its own data...”; column 7, lines 6-46 “...The two threads **16** and **18** then will operate essentially independently of each other, reading and executing different instructions and using different data...”; column 11, line 37 to column 12, line 4 “...a DFORK instruction causing instruction pipeline **2** to being operating in MIMD mode at time t_2 ...”; Figure 4; Figure 5; and Figure 7); and

c. In the wide mode:

- i. Fetching a fifth bundle of instructions from a fifth thread of the program (Fernando column 3, lines 47-61 “In the SIMD mode, the fetch stage, decoder stage, and execute stage of the first of the parallel instruction pipelines is fully operational. One or more of the decoder stages of the other instruction pipelines are coupled to received the instructions fetched by the fetch stage of the first instruction pipeline...”; column 6, lines 33-54 “...Secondary instruction pipeline **18** then commences to execute the code fetched by fetch stage **20a**, while primary instruction pipeline **16**

continues to execute the same instructions.”; column 11, line 37 to column 12, line 4 “...At time t₉, a CFORK instruction is executed in instruction pipeline 1 activating instruction pipeline 2 in the SIMD mode...”; Figure 3; Figure 5; and Figure 7);

- ii. Distributing the fifth bundle to the first cluster for execution therethrough (Fernando column 3, lines 47-61 “In the SIMD mode, the fetch stage, decoder stage, and execute stage of the first of the parallel instruction pipelines is fully operational. One or more of the decoder stages of the other instruction pipelines are coupled to received the instructions fetched by the fetch stage of the first instruction pipeline...”; column 6, lines 33-54 “...Secondary instruction pipeline 18 then commences to execute the code fetched by fetch stage 20a, while primary instruction pipeline 16 continues to execute the same instructions.”; column 11, line 37 to column 12, line 4 “...At time t₉, a CFORK instruction is executed in instruction pipeline 1 activating instruction pipeline 2 in the SIMD mode...”; Figure 3; Figure 5; and Figure 7);
- iii. Fetching a sixth bundle of instructions from the fifth thread of the program (Fernando column 3, lines 47-61 “In the SIMD mode, the fetch stage, decoder stage, and execute stage of the first of the parallel instruction pipelines is fully operational. One or more of the decoder stages of the other instruction pipelines are coupled to received the instructions fetched by the fetch stage of the first instruction pipeline...”; column 6, lines 33-

54 "...Secondary instruction pipeline 18 then commences to execute the code fetched by fetch stage 20a, while primary instruction pipeline 16 continues to execute the same instructions."; column 11, line 37 to column 12, line 4 "...At time t9, a CFORK instruction is executed in instruction pipeline 1 activating instruction pipeline 2 in the SIMD mode..."; Figure 3; Figure 5; and Figure 7 -- In regards to Fernando, in SIMD mode, the two instruction pipelines execute instructions from the same thread, so all bundles fetched from this thread, including the third and fourth bundles, are executed in both the first and second instruction pipelines. Also, there is nothing in the claim language stating that the fourth bundle of instructions is different from the third bundle of instructions. In addition, Fernando has taught in column 1, line 54 to column 2, line 3 the use of VLIW instructions where each VLIW instruction fetch contains multiple single operation instructions for parallel execution in a datapath such as those shown in Figure 1.); and

- iv. Distributing the sixth bundle to the second cluster for execution therethrough (Fernando column 3, lines 47-61 "In the SIMD mode, the fetch stage, decoder stage, and execute stage of the first of the parallel instruction pipelines is fully operational. One or more of the decoder stages of the other instruction pipelines are coupled to received the instructions fetched by the fetch stage of the first instruction pipeline..."; column 6, lines 33-54 "...Secondary instruction pipeline 18 then

commences to execute the code fetched by fetch stage **20a**, while primary instruction pipeline **16** continues to execute the same instructions.”; column 11, line 37 to column 12, line 4 “...At time **t9**, a CFORK instruction is executed in instruction pipeline **1** activating instruction pipeline **2** in the SIMD mode...”; Figure 3; Figure 5; and Figure 7).

6. Referring to claim 20, Fernando has taught the method of claim 19, further comprising:
 - a. Processing the first, second, and fifth bundles within the first cluster (Fernando column 11, line 37 to column 12, line 4 “...instruction pipelines **1**, **2** and **3** operate independently of each other in MIMD mode...instruction pipeline **2** starts to accept instructions from the fetch stage in instruction pipeline **1**...” and Figure 7);
 - b. Processing the third, fourth, and sixth bundles within the second cluster (Fernando column 11, line 37 to column 12, line 4 “...instruction pipelines **1**, **2** and **3** operate independently of each other in MIMD mode...instruction pipeline **2** starts to accept instructions from the fetch stage in instruction pipeline **1**...” and Figure 7).
7. Referring to claim 21, Fernando has taught the method of claim 19, further comprising the step of architecting data from the first cluster to a first register file (Fernando column 4, line 57 to column 5, line 3 “...There is a register **25** for each of the execute stages...for temporarily storing data...data to be operated on by the instructions. The register files of all of the pipelines...” and Figure 1).
8. Referring to claim 22, Fernando has taught the method of claim 19, further comprising the step of architecting data from the second cluster to a second register file (Fernando column 4, line 57 to column 5, line 3 “...There is a register **25** for each of the execute stages...for

temporarily storing data...data to be operated on by the instructions. The register files of all of the pipelines...” and Figure 1).

9. Referring to claim 23, Fernando has taught The method of claim 19, the step of fetching the first bundle comprising decoding instructions into the first bundle (Fernando column 1, line 54 to column 2, line 3 “...code is either written or compiled to cause such independent instructions to be grouped into a VLIW. Each VLIW is parsed and then fed into multiple issue slots in the processor...”; column 4, lines 46-49 “Decoder stages...”; and Figure 1).

10. Referring to claim 27, Fernando has taught the method of claim 19, further comprising bypassing data between the first cluster and the second cluster, as needed, to facilitate the processing of the fifth bundle through the first cluster and the sixth bundle through the second cluster (Fernando column 4, line 64 to column 5, line 3 “...data may be exchanged) responsive to the appropriate instruction or instructions) between the register file **25a** of execute stage **24a** of the primary instruction pipeline **16** and the register files **25b** of the execute stage **24b** of the secondary instruction pipeline **18** via a register bus **38**...” and Figure 1, element 38).

11. Referring to claim 29, Fernando has taught the method of claim 19, wherein the step of determining the mode of operation comprises determining a state of a configuration bit (Fernando column 2, lines 51-62 “...The architecture is capable of running in various modes, including single threaded mode, SIMD mode and MIM mode...”; column 3, line 29-33 “...at least three different modes of parallelism...”; column 6, lines 33-40 “...CFORK instruction is one of the special instructions and, specifically, is the instruction which activates one or more secondary instruction pipelines in the SIMD mode...”; column 7, lines 14-35 “...DFORK is one of the special instructions. This instruction indicates that the architecture is to enter the MIMD

mode...”; and Figure 5 – In regards to Fernando, the instructions are comprised of bits that signal when the system is in SIMD or MIMD modes.).

12. Referring to claim 31, claim 31 contains similar limitations to those found in claims 19 and 29 above and is rejected for similar reasons. The only difference between claim 31 and claims 19 and 29 is that claim 31 is a processor claim while claims 19 and 29 are method claims.

13. Referring to claim 32, claim 32 contains similar limitations to claims 21 and 22 above and is rejected for similar reasons. The only difference between claim 32 and claims 21 and 22 is that claim 32 is a processor claim while claims 21 and 22 are method claims.

14. Referring to claim 34, claim 34 contains similar limitations to claims 20 and 30 above and is rejected for similar reasons. The only difference between claim 34 and claims 20 and 30 is that claim 34 is a processor claim while claims 20 and 30 are method claims.

Claim Rejections - 35 USC § 103

15. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

16. Claims 28 and 33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fernando et al., U.S. Patent Number 6,272,616 (herein referred to as Fernando) in view of Higuchi et al., U.S. Patent Number 5,303,354 (herein referred to as Higuchi).

17. Referring to claims 28 and 33, Fernando has taught wherein the step of bypassing the data between a register file of the first cluster and a register file of the second cluster (Fernando column 4, line 64 to column 5, line 3 “...data may be exchanged (responsive to the appropriate

Art Unit: 2183

instruction or instructions)...”). Fernando has not explicitly utilizing a latch to couple the data between the registers. Higuchi has explicitly taught utilizing a latch to couple the data between registers when bypassing data from register to register (Higuchi Abstract “Data is transferred directly from a source register in a register file connected to a data bus from a destination register in the register file, through a read data latch...”; column 2, lines 36-45 “...there are provided a read data latch **21** for latching data read from a desired register...”; column 3, line 49 to column 4, line 26 “...The read data latch **21** reads data from a desired register...”; and Figure 1). A person of ordinary skill in the art at the time the invention was made, and as taught by Higuchi, would have recognized that a bypass using the latch eliminates the need for the ALU or processing unit during a data transfer, thus speeding up data transfer between registers and freeing up resources for other operations (Higuchi column 2, lines 27-36 “...thus making it possible to speed up data transfer...”). Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made that incorporating the bypass with a latch of Higuchi in the device of Fernando increases data transfer speed and frees up resources for other operations.

Response to Arguments

18. Applicant's arguments with respect to claims 28 and 33 have been considered but are moot in view of the new ground(s) of rejection.

19. Applicant's arguments filed 09 July 2007 have been fully considered but they are not persuasive. Applicant argues in essence on pages 5-7 with regards to claims 19-23, 27, 29, and 31-32

...Thus, the Applicant contends that Fernando does not teach or suggest two clusters, each of which receive instruction bundles from two separate program threads while in throughput mode...

20. This has not been found persuasive. Applicant's arguments appear to focus on the fact that Fernando's examples of their invention only mention one main program with one fork thread operating per pipeline and insinuating that, since only one main program with one fork thread operating per pipeline is shown in Fernando, there will be only one main program ever run on the device and each pipeline will run only one fork thread. However, it is extremely common for resources to be re-used within a system when the resource, such as a pipeline, is free. This is even illustrated in Fernando's Figure 7 when the second pipeline is re-used at time 10-12 when the system is operating in SIMD mode. Data Path #2, which had been used previously in MIMD mode to execute thread 2 but is free at time 10, is re-used in SIMD mode by the main pipeline. As such, the pipelines will process any number of threads, as long as they are free. Thus, the individual pipelines, i.e. clusters, will receive multiple separate program fork threads for execution. For example, with regards to Fernando's Figure 7, Data Path #1 executes the main thread, so, when the current main thread completes, another main thread will be run. It stands to reason that the second main thread will be run on Data Path #1, since it is free and the resources associated with it are available for use. The second main thread, if in MIMD mode, will spawn fork threads that are executed by Data Path # 2, 3, and 4, or, if in SIMD mode, Data Path #2 will also execute instructions from the second main thread. This means that the Data Paths execute instructions not only from the main thread and its fork threads, as shown in Figure 7, and from the second main thread and its fork threads. The exemplary embodiments of Fernando are

Art Unit: 2183

exactly that, exemplary embodiments. They are meant for simple explanation of how Fernando's system function, and not limiting Fernando's system to executing only a single main program and three fork threads on four pipelines. To assume that one main thread, or program, with four forks are the only threads run on the system is contrary to accepted practices and knowledge in the art.

Conclusion

21. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

22. A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

23. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Aimee J. Li whose telephone number is (571) 272-4169. The examiner can normally be reached on M-T 7:00am-4:30pm.

Art Unit: 2183

24. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

25. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.



Aimee J Li
Examiner
Art Unit 2183

13 September 2007